

CLAIMS

Amend the claims as follows.

1. (Currently amended) A database management system, comprising:
a processor configured to:

associate multiple different activities with a same transaction, each of the multiple different activities each consisting of a separate different associated subgroup of program instructions for the same transaction,

for each of the different subgroups of program instructions, ~~that~~ initiate a different associated subgroup of multiple different read and/or write actions that access an associated group of multiple different data items;

use and assign, for each of the different activities, only one single separate lock duration for all of the multiple different data items associated with ~~each the~~ different associated subgroup of program instructions ~~associated with each of the different activities~~;

maintain, for each of the different activities, ~~the~~ multiple different locks on all of the multiple different data items associated with the ~~same activities~~ activity, and then ~~releasing~~ release all of the multiple different locks for all of the multiple different data items associated with the ~~same activities~~ activity together only when all of the subgroup of program instructions associated with the ~~same activities~~ activity are completed so that all of the multiple different locks on all of the multiple different data items associated with the ~~same activities~~ activity have a same lock duration;

release all of the locks on a first ~~set~~ one of the groups of ~~the~~ multiple different data items associated with a first one of the different activity activities of the same transaction while a second set of data items that include at least some of the first ~~set~~ group of data items from the first activity, but that are associated with a second one of the different activity activities for the same transaction, remain locked for a second one of the separate single lock durations associated with a the second activity.

2. (Currently amended) A database management system according to claim 1 wherein one of the activities includes a group of individual shared lock operations and the

processor activates locks for each of the individual shared lock operations in the group and releases the locks only when the all of the individual shared lock operations in the group are completed.

3. (Original) A database management system according to claim 1 including a memory containing an activity bit map that tracks individual activities for the transaction, the processor assigning activity identifiers to the activities according to the activity bit map.

4. (Currently amended) A database management system according to claim 1 wherein, for each of the different activities, the processor releases all of the multiple different locks associated with the ~~same activities~~ activity in one operation only when ~~all of the multiple different~~ subgroup of program instructions associated with the ~~same activities are~~ activity is completed.

5. (Previously Presented) A method for locking data items in a database management system, comprising:

associating multiple different activities with a same transaction, each of the activities consisting of a different associated subgroup of program instructions for the same transaction and each different subgroup of program instructions initiating a different associated subgroup of read and/or write actions accessing an associated group of multiple different data items;

associating and using a single separate individual lock duration for each different subgroup of program instructions and the accessed multiple different data items associated with the same activities in the same transaction;

maintaining multiple different locks on all of the multiple different data items for each different subgroup of program instructions associated with the same activities; and

releasing all of the multiple different locks for all of the multiple different data items associated with the same activities together only when the entire subgroup of program instructions associated with the same activities are all completed.

6. (Currently amended) A method according to claim 5

further comprising releasing all of the locks on a first ~~set~~ one of the groups of multiple different data items associated with a first one of the activity activities of the transaction while a second set of data items that include at least some of the first ~~set~~ group of data items from the first activity, but that are also associated with a second one of the activity activities for the same transaction, remain locked for a ~~the~~ separate single individual lock duration associated with the second activity.

7. (Original) A method according to claim 5 including:

maintaining an activity bit map that tracks individual activities for the transaction; and
assigning activity identifiers to the activities according to the activity bit map.

8. (Previously Presented) A method according to claim 6 including associating the activities with lock modes and releasing the lock modes on data items when the associated activities are finished.

9. (Currently amended) The method according to claim 5 including:

assigning a same unique activity identifier to multiple different arbitrary database access instructions that constitute one of the different activities in the transaction, the database access instructions performing one or more operations on multiple different data items in a database and the activity identifier assigned to and associated with the database access instructions independently of any relationship that may exist between the multiple different data items in the database accessed by the database access instructions;

assigning multiple locks to the multiple data items corresponding with the operations performed on the multiple data items pursuant to the database access instructions; and

preventing other transactions and other associated activities from accessing the multiple different data items until all of the ~~multiple~~ operations are completed for all of the database access instructions assigned to the activity identifier.

10. (Currently amended) A database management system, comprising:

a processor configured to assign activity identifiers to different individual subgroups of database access instructions for a same transaction that each perform one or more operations on

multiple data items in a database, the activity identifiers assigned to and associated with the database access instructions independently of any relationship that may exist between the multiple data items in the database accessed by the database access instructions,

the processor further configured, for each of the assigned activity identifiers, to assign multiple locks to the multiple data items corresponding with the operations performed on the multiple data items pursuant to the database access instructions associated with the ~~same~~ assigned activity identifiers identifier, and further configured to only release the multiple locks on the multiple data items when all of the ~~multiple~~ operations are completed for all of the database access instructions assigned to the ~~same assigned~~ activity identifiers identifier, and wherein the release of the multiple locks is in response to a single request via the assigned activity identifier.

11. (Previously Presented) The database management system according to claim 10 wherein the processor is further configured to assign the activity identifiers to an arbitrary group of related database access instructions performing operations on an arbitrarily related group of data items.

12. (Previously Presented) The database management system according to claim 10 wherein the processor is further configured to assign common transaction identifiers to different related groups of database access instructions assigned different activity identifiers and coordinate when the different related groups of database access instructions are allowed to perform operations on the data items.

13. (Previously Presented) A database management system according to claim 10 wherein the processor is configured to assign a first transaction identifier to a group of individual shared operations and assign locks to the data items associated with the shared operations, the processor further configured to hold the locks until all of the individual shared operations in the group have been completed.

14. (Currently amended) Computer readable media containing instructions that when executed by a computer, comprise:

assigning activity identifiers to different individual subgroups of database access instructions within a same transaction that perform multiple operations on multiple different data items in a database, the activity identifiers assignable to the database access instructions independently of any relationship that may exist between the multiple data items in the database accessed by the database access instructions;

assigning multiple locks to the multiple different data items corresponding with the operations performed on the multiple data items by the different subgroups of database access instructions;

for each of the different subgroups of database access instructions, assigning the same one of the activity identifiers to the locks that are associated with the same subgroups subgroup of database access instructions; and

releasing, for each of the assigned activity identifiers, all of the multiple locks on all of the multiple different data items assigned to the same assigned activity identifiers identifier at the same time when all of the multiple operations are completed for all of the subgroups subgroup of database access instructions assigned to the same assigned activity identifiers identifier, and wherein the releasing of all of the multiple locks is in response to a single request via the assigned activity identifier.

15. (Previously Presented) The computer readable media according to claim 14 including instructions that when executed assign the activity identifiers to an arbitrary group of related database access instructions performing operations on an arbitrarily related group of data items.

16. (Previously Presented) The computer readable media according to claim 14 including instructions that when executed assign common transaction identifiers to different related groups of database access instructions each assigned different activity identifiers and coordinate when the related groups of database access instructions are allowed to perform operations on the data items.

17. (Previously Presented) The computer readable media according to claim 14 including instructions that when executed assign a first transaction identifier to a group of

individual shared operations, assign locks to the data items associated with the shared operations, and hold the locks until all of the individual shared operations in the group have been completed.

18. (Currently amended) A database management system, comprising:

means for associating multiple different activities with a same transaction, each of the activities consisting of an associated subgroup of program instructions for the same transaction that initiate a subgroup of actions on an associated group of multiple different data items;

means for associating separate sets of locks with each of the different activities in the transaction;

means for maintaining the separate sets of locks for the duration of the different activities; and

means for releasing, for each of the different activities, the separate sets of locks on the ~~entire~~ associated group of multiple different data items at the same time when the associated ~~subgroups~~ subgroup of program instructions ~~associated with the activities are~~ is completed.

19. (Previously Presented) The database management system according to claim 18 including:

means for identifying a plurality of different activities for the same transaction;

means for activating locks on data items associated with the activities; and

means for releasing the locks on the data items when the associated activities are completed.

20. (Currently amended) The database management system according to claim 18 including:

means for maintaining an activity bit map that tracks individual activities for the ~~transaction associated with a~~ same transaction; and

means for assigning activity identifiers to the activities according to the activity bit map.

21. (Previously Presented) The database management system according to claim 18 including means for identifying one or more subclasses of activities within an activity and

associating lock durations with the subclass and releasing the locks upon completion of the activities in the subclass before releasing the locks on the activity.

22. (Currently amended) The database management system according to claim 10 wherein the processor is configured to assign the locks associated transaction identifiers and associated ones of the activity identifiers and to release ~~a groups~~ group of the locks only when all of the ~~multiple~~ operations are completed for all of the database access instructions having the a same assigned one of the transaction identifiers and a same assigned one of the activity identifiers as the group of the locks.

23. (Currently amended) A method, comprising:

assigning a first activity identifier and a transaction identifier to a first group of database access instructions for a transaction;

assigning a first set of multiple locks to a first set of multiple different data items accessed by the first group of database access instructions, the multiple locks assigned to different ones of the first set of data items according to the first ~~sub~~group of database access instructions;

identifying a second subset of multiple different data items from the first set of data items ~~aeording to~~ accessed by the first group of database access instructions;

assigning a second activity identifier and the same transaction identifier to a second group of database access instructions for the same transaction that modify the second subset of multiple different data items ~~identified by the first group of database access instructions~~;

assigning a second set of multiple locks to the second subset of multiple different data items, the second set of multiple locks having a different lock duration than the first set of multiple locks;

releasing the first set of multiple locks only for the first set of data items that are not part of the second subset of multiple different data items only after the ~~second~~ first group of database access instructions have completed; and

releasing the entire second set of multiple locks via a single request to release all locks associated with the second activity identifier only when all of the operations for the second group

of database access instructions have completed modification of the second subset of multiple different data items.

24. (Previously Presented) The method according to claim 23 including releasing all of the first set of locks in one operation and releasing all of the second set of locks in one operation.